

Baysian Super-resolved Surface Reconstruction from multiple images

V. N. Smelyanskiy, P. Cheesman, D.A. Maluf, R.D. Morris

RIACS

Moffet Field, California 94035-1000, USA

Bayesian Super-resolved Surface Reconstruction from multiple images

Abstract

Bayesian inference has been used successfully for many problems where the aim is to infer the parameters of a model of interest. In this paper we formulate the three dimensional reconstruction problem as the problem of inferring the parameters of a surface model from image data, and show how Bayesian methods can be used to estimate the parameters of this model given the image data. Thus we recover the three dimensional description of the scene. This approach also gives great flexibility. We can specify the geometrical properties of the model to suit our purpose, and can also use different models for how the surface reflects the light incident upon it. In common with other Bayesian inference problems, the estimation methodology requires that we can simulate the data that would have been recorded for any values of the model parameters. In this application this means that if we have image data we must be able to render the surface model. However it also means that we can infer the parameters of a model whose resolution can be chosen irrespective of the resolution of the images, and may be super-resolved. We present results of the inference of surface models from simulated aerial photographs for the case of super-resolution, where many surface elements project into a single pixel in the low-resolution images.

1 Introduction

Bayesian inference has proved to be the method of choice for many inference problems enabling accurate estimation of parameters of interest from noisy and incomplete data, and also providing estimates of the errors associated with the inferred parameters. The general approach is illustrated in figure 1. The figure shows that synthetic observations of the model are made using a computer simulation of the observation process, and these are compared with the actual observations. The error between the actual and the simulated observations is used to adjust the parameters of the model, to minimize the errors. The estimation of the errors on the parameter estimates (or more generally, the estimation of the covariance matrix of the parameters) means that the parameter estimates can be updated when more observations become available.

In the application to the reconstruction of three dimensional surfaces, the model is the parameterized surface model and the measurement system is a computer model of the image formation process.

The surface model is chosen to suit the desired application. For example it could be a CAD model if the data was images of machine parts, or a spline model if the data came from more general man-made objects. In the application we describe in this paper we are interested in recovering planetary surfaces, and hence we choose to describe the surface by a triangular mesh. This is a very standard model in computer graphics, but is little used in computer vision (a notable exception is [?] (zisserman)). There is also the flexibility to specify the density of the mesh, which need not be spatially constant.

The computer model of the image formation process is the area of computer graphics known as rendering. A model of how light is reflected from the surface is used, together with an abstraction of how the image is formed, to synthesize the image that would have been recorded from the current surface under the lighting conditions and camera position and orientation. These light reflectance parameters are parameters of the surface model that are to be inferred together with the geometrical parameters. The theoretical exposition in this paper is valid for any parameterized reflection function. The results assume Lambertian reflection.

The formation of the synthetic image from the parameterized model and the lighting and camera parameters is discussed in more detail in section 3. Here we just note that current rendering technology is unsuitable for our purpose because it operates in *image space*, and so the image formed has artifacts due to the relative sizes of the projection of the surface elements onto the image plane and the pixels. These artifacts are noticeable along the edges of surface elements. Also, if the projections of the surface elements in the pixel plane are very small, such as when we are trying to infer a high resolution mesh from many low resolution images, these artifacts dominate the 'standard' rendering process. The renderer that is required for this purpose is one that operates in *object space*,

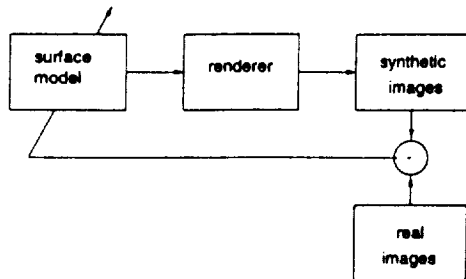


Figure 1: An outline of the Bayesian approach to surface reconstruction

and in section 3 we briefly describe such a system. An object-space renderer can also be made to compute the *derivatives* of the pixel values with respect to the surface model parameters. This is a crucial part of the inference process, and is described in detail in section 3.1.

One of the major goals of this paper is to develop Bayesian inference approach for 3D super-resolved surface reconstruction when the resolution of inferred surface mesh is higher than the spatial resolution of input images. This also allows to achieve the image super-resolution: synthetically produced images of the super-resolved surface model can be at higher resolution than that of input data images.

1.1 Previous Work

Most previous work in the area of the estimation of three dimensional surfaces from image data has used one of two main approaches, shape from shading (more generally, 'shape from X') and shape from stereo.

In shape from shading [?] the image gradients are used, and, using the assumption of orthographic projection, are related to the surface derivatives. Using assumed boundary conditions the surface derivatives can be integrated to produce an estimate of the surface heights (more strictly, the distance from the camera to the points on the surface).

Shape from stereo [?] uses correspondence matches between features in the images to give the disparity between these features. Finding these correspondence matches is aided by using the epipolar constraint [?]. If the camera geometry is known then the disparity can be directly related to the distance of the feature from the camera, and the feature can be located in space. The discrete points corresponding to the features matched in the images are then joined to form a representation of the surface.

A drawback of shape from stereo is that the density of points in the recovered surface is unknown a-priori, and is dependent on the number and density of fea-

tures found in the images. A feature detector giving few features gives a very coarse surface representation; one that gives very many is likely to produce features that are hard to match between images.

Shape from shading has the drawback that the spatial density of the recovered surface is fixed at the image density. It is also difficult to apply if the reflectance properties are spatially varying. Both of these approaches have difficulty incorporating new observations of the surface that become available after the initial estimate is made.

Bayesian approach was used for image super-resolution in the number of papers beginning from [6]. In the previous work [7] input images were taken from roughly the same direction under the similar lightening conditions. In this work the surface model is essentially represented as 2D texture map. In our case of fully 3D surface reconstruction this restriction is lifted: low resolution input images as well as high-resolution output images can correspond to a very different values of registration parameters.

2 A Bayesian Framework

In this paper we consider the following surface model. The geometry is represented by a triangular mesh. We consider the case of Lambertian surfaces and store surface reflection properties in the vector of albedo values associated with the vertices of the triangular mesh. We will assume known the camera parameters and the parameters of the lighting. The estimation of these parameters will be considered in a forthcoming paper using the same Bayesian framework.

Thus we represent the surface model by the pair of vectors $[h, \rho]$. The components of these vectors correspond to the height and albedo values defined on a regular grid of points

$$[h\rho] = \{(z_i, \rho_i), \quad i = \ell(q\hat{x} + p\hat{y})\} \quad q, p = 0, 1, \dots \quad (1)$$

where ℓ elementary grid length, \hat{x}, \hat{y} are an orthonormal pair of unit vectors in the (x, y) plane and i indexes the position in the grid. The pair of vectors of heights and albedos represent a full vector for the surface model

$$u = [h \ \rho]. \quad (2)$$

To estimate the values of h, ρ from image data, we apply Bayes theorem which gives

$$p(h, \rho | I_1 \dots I_F) \propto p(I_1 \dots I_F | h, \rho) p(h, \rho)$$

where I_f ($f = 1, \dots, F$) is the image data. This states that the posterior distribution of the heights and the

albedos is proportional to the likelihood – the probability of observing the data given the heights and albedos – multiplied by the prior distribution on the heights and albedos.

The prior distribution is assumed to be Gaussian

$$p(h, \rho) \propto \exp\left(-\frac{1}{2}\mathbf{u}\Sigma^{-1}\mathbf{u}^T\right), \quad (3)$$

$$\Sigma^{-1} = \begin{bmatrix} \hat{Q}/\sigma_h^2 & 0 \\ 0 & \hat{Q}/\sigma_\rho^2 \end{bmatrix},$$

where the vector of the surface model parameters \mathbf{u} is defined in (2). The inverse covariance matrix is constructed to enforce the smoothing constraint on local variations of heights and albedos. We penalize the integral over the surface of the square of the surface curvature $c(x, y) = h_{xx}^2 + h_{yy}^2 + 2h_{xy}^2$, and similarly for albedos. We approximate the partial derivatives in $c(x, y)$ using finite differences of the height (albedo) values. Then coefficients of \hat{Q} form a 5×5 template Γ and result from summing $c(x, y)$ over the surface.

$$Q_{n,m}^{q+n,p+m} = \Gamma_{q,p}, \quad q, p = -2, \dots, 2. \quad (4)$$

For brevity we do not provide here an explicit values of coefficients $\Gamma_{q,p}$. Two metaparameters σ_h and σ_ρ in equation (3) control the expected values of the surface-averaged curvatures for heights and albedos.

This prior is placed directly over the height variables, h , but albedos are only defined over the range $[0 - 1]$. Because of this we put in equation (3) the Gaussian prior for the albedos over transformed variables, where

$$\rho \rightarrow \log(\rho'/(1 - \rho')), \quad \mathbf{u} = [h \ \rho'] \quad (5)$$

In the vector of model parameters \mathbf{u} values of ρ are replaced by values of ρ' .

For the likelihood we make the usual assumption that the differences between the observed data and the data synthesized from the model have a zero mean, Gaussian distribution, and also assume that the images I_f comprising the data are conditionally independent. This gives

$$p(I_1 \dots I_F | h, \rho) \propto \exp\left(-\frac{\sum_{f,p}(I_{fp} - \hat{I}_{fp}(h, \rho))^2}{2\sigma_e^2}\right)$$

where $\hat{I}_{fp}(h, \rho)$ denotes the pixel intensities in the image f synthesized from the model, σ_e^2 is the noise variance and the summation is over the pixels (p) and over all images (f) used for the inference.

Consider the negative log-posterior.

$$L(h, \rho) \propto \frac{\sum_{f,p}(I_{fp} - \hat{I}_{fp}(h, \rho))^2}{\sigma_e^2} + \mathbf{u}\Sigma^{-1}\mathbf{u}^T \quad (6)$$

where the model parameters vector is defined in (2). This is a nonlinear function of h, ρ and the MAP estimate is that value of h, ρ which minimizes $L(h, \rho)$.

In the case of images with no shadows or visible occlusions which we consider here, the log-posterior is in general unimodal and gradient methods can be applied for minimizing $L(h, \rho)$. We linearize $\hat{I}(h, \rho)$ about the current estimate, h_0, ρ_0

$$\hat{I}(h, \rho) = \hat{I}(h_0, \rho_0) + \mathbf{D}(\mathbf{u} - \mathbf{u}_0), \quad \mathbf{D} \equiv \left\{ \frac{\partial \hat{I}_{fp}}{\partial z_i}, \frac{\partial \hat{I}_{fp}}{\partial \rho'_i} \right\} \quad (7)$$

where \mathbf{D} is the matrix of derivatives evaluated at h_0, ρ_0 . Then minimization of $L(h, \rho)$ is replaced by minimization of the quadratic form:

$$L' = \frac{1}{2}\mathbf{x}\hat{A}\mathbf{x} - \mathbf{b}\mathbf{x}, \quad \mathbf{x} \equiv \mathbf{u} - \mathbf{u}_0,$$

$$\hat{A} = \Sigma^{-1} + \frac{\mathbf{D}\mathbf{D}^T}{\sigma_e^2}, \quad (8)$$

$$\mathbf{b} = \frac{(I - \hat{I}(h_0, \rho_0))}{\sigma_e^2}\mathbf{D} + \Sigma^{-1}\mathbf{u}_0.$$

Here \hat{A} is the Hessian matrix of the quadratic form and vector \mathbf{b} is a gradient of a likelihood L computed at current estimate. We search for the minimum in \mathbf{x} using a conjugate-gradient method. At the minimum we update the current estimate, $\mathbf{u}_1 = \mathbf{u}_0 + \mathbf{x}$, recompute \hat{I} and \mathbf{D} , and repeat the minimization procedure iteratively until the current estimate \mathbf{u}_k approaches a global minimum of $L(h, \rho)$.

Thus to find the MAP estimate requires that we can render the image and compute the derivatives for any values of the surface model parameters. We discuss this computation in some detail in the next sections. Here it is sufficient to note that while forming \hat{I} using only object space computation (see section 3) is computationally expensive, we can compute \mathbf{D} at the same time for little additional computation. Also the derivative matrix is sparse with the number of nonzero entries a few times the number of model parameters. This makes the process described above a practical one. Convergence is also accelerated by using a multi-grid approach.

At convergence we compute a new inverse covariance matrix, $(\Sigma^{-1})' = \Sigma^{-1} + \mathbf{D}\mathbf{D}^T/\sigma_e^2$. This is then used as a *prior* inverse covariance matrix when new image data of the same surface is obtained, enabling

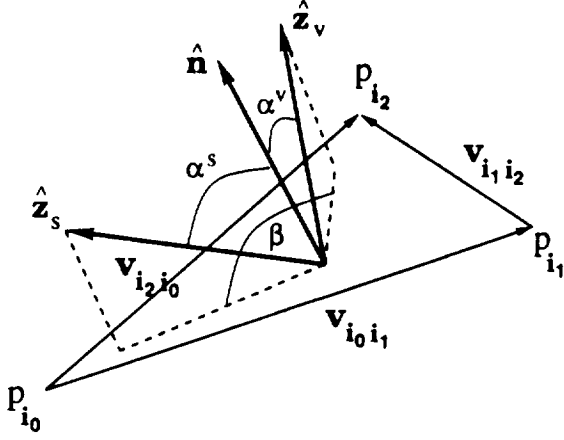


Figure 2: Geometry of the triangular facet, illumination direction and viewing direction

a recursive update and integration of data recorded at different times. The posterior inverse covariance matrix gives information about the uncertainty of the estimated surface.

3 Formation of the image and the derivative matrix.

The task of forming an image, \hat{I} , given a surface description, h, ρ , and camera and illumination parameters is the area of computer graphics known as rendering[2]. Most current rendering technology is focused on producing images which are *visually* appealing, and producing them very quickly. As discussed in the introduction, this results in the use of image-space algorithms, with the fundamental assumption that each triangle making up the surface, when projected onto the image plane, is much larger than a pixel. This makes reasonable the assumption that any given pixels receives light from only one triangle, but does produce images with artifacts at the triangle edges. This approach also produces inaccurate images if the triangles project into areas much smaller than a pixel on the image plane, as the pixel will then be colored with a value coming from just *one* of the triangles.

Clearly this approach is not suitable for high-resolution 3D surface reconstruction from multiple images. The triangles in a high-resolution surface may project onto an area much smaller than a single pixel in the image plane (sub-pixel resolution). Therefore, as discussed in the introduction, for our system we implemented a renderer for triangular meshes which performs all computation in *object space*. At present we neglect the blurring effect due to diffraction and

due to the role of pixel boundaries in the CCD array. Then the light from a triangle as it is projected into a pixel contributes to the brightness of the pixel with a weight factor proportional to the fraction of the area of the triangle which projects into that pixel. This produces perfectly anti-aliased images and allows an image of any resolution to be produced from a mesh of arbitrary density, as required when the system performing the surface inference may have no control over the image data gathering.

Our renderer computes brightness \hat{I}_p of a pixel p in the image as a sum of contributions from individual surface triangles Δ whose projections into the image plane overlap, at least partially, with the pixel p .

$$\hat{I}_p = \sum_{\Delta} f_{\Delta}^p \Phi_{\Delta}. \quad (9)$$

Here Φ_{Δ} is a radiation flux reflected from the triangular facet Δ and received by the camera, and f_{Δ}^p is the fraction of the flux that falls onto a given pixel p in the image plane. In the case of Lambertian surfaces and single spectral band Φ_{Δ} is given by the expression

$$\Phi_{\Delta} = \rho E(\alpha^s) \cos \alpha^v \cos^{\kappa} \theta \Delta\Omega, \quad (10)$$

$$E(\alpha^s) = \rho \mathcal{A} (I^s \cos \alpha^s + I^a).$$

$$\Delta\Omega = S/d^2.$$

Here ρ is an average albedo of the triangular facet. Orientation angles α^s and α^v are defined in figure 2. $E(\alpha^s)$ is the total radiation flux incident on the triangular facet with area \mathcal{A} . This flux is modeled as a sum of two terms. The first term corresponds to direct radiation with intensity I^s from the light source at infinity (commonly the sun). The second term corresponds to ambient light with intensity I^a . The parameter θ in equation. (10) is the angle between the camera axis and the viewing direction (the vector from the surface to the camera); κ is the lens falloff factor. $\Delta\Omega$ in (10) is the spatial angle subtended by the camera which is determined by the area of the lens S and the distance d from the centroid of the triangular facet to the camera.

We identify the triangular facet Δ by the set of 3 indices (i_0, i_1, i_2) from the vector of heights (1) that determines the vertices of the triangle in a counter-clockwise direction (see figure 2). In the r.h.s of equation (10) we have omitted for brevity those indices from all the quantities associated with individual triangles. The average value of albedo for triangle in (10) is computed based on the components of the albedo vector ρ corresponding to the triangle indices

$$\rho_{\Delta} \equiv \rho_{i_0, i_1, i_2} = \frac{1}{3} (\rho_{i_0} + \rho_{i_1} + \rho_{i_2}). \quad (11)$$

We note that using average albedo (11) in the expression for Φ_Δ is an approximation which is justified when the albedo values vary smoothly between the neighboring vertices of a grid.

The area \mathcal{A} of the triangle and the orientation angles in (10) can be calculated in terms of the vertices of the triangle \mathbf{P}_i (see figure 2) as follows:

$$\begin{aligned} \hat{\mathbf{n}} \cdot \hat{\mathbf{z}}^x &= \cos \alpha^x, \quad \hat{\mathbf{n}} \cdot \hat{\mathbf{z}}^y = \cos \alpha^y, \\ \hat{\mathbf{n}} &= \frac{\mathbf{v}_{i_0, i_1} \times \mathbf{v}_{i_1, i_2}}{2\mathcal{A}}, \quad \mathbf{v}_{i,j} = \mathbf{P}_j - \mathbf{P}_i \end{aligned} \quad (12)$$

Here $\hat{\mathbf{n}}$ is a unit normal to the triangular facet and vectors of the edges of the triangle $\mathbf{v}_{i,j}$ are shown in figure 2.

We use a standard pinhole camera model with no distortion in which coordinates of a 3D world point $\mathbf{P} = (x, y, z)$ are first rotated with the rotation matrix $\hat{\mathbf{R}}$ and translated by the vector \mathbf{T} into camera coordinates, yielding $\mathbf{P}_c = (x_c, y_c, z_c)$

$$\mathbf{P}_c = \hat{\mathbf{R}} \mathbf{P} + \mathbf{T}. \quad (13)$$

($\hat{\mathbf{R}}$ and \mathbf{T} are expressed in terms of the camera registration parameters [?]; we do not give them explicitly here). After the 3D transformation given in (13) point \mathbf{P}_c in the camera coordinate system is transformed using a perspective projection into the 2D image point $\bar{\mathbf{P}} = (\bar{x}, \bar{y})$ using a focal length f and aspect ratio a .

$$\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = -\frac{f}{z_c} \begin{bmatrix} a & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \end{bmatrix}. \quad (14)$$

We use 2D image projections of the triangular vertices \mathbf{P}_i to compute the area fraction factors f_Δ^p for surface triangles (cf. Eq. (9))

$$f_\Delta^p = \frac{\bar{A}_{\text{polygon}}}{\bar{A}_\Delta}. \quad (15)$$

Here \bar{A}_Δ is the area of the projected triangle on the image plane and \bar{A}_{polygon} is the area of the polygon resulting from the intersection of the projected triangle and boundary of the pixel p (see figure 3).

Thus, our rendering algorithm does the following:

1. Transform the coordinates of the vertices of the surface model, \mathbf{P}_i , into 2D image coordinates. (Maintaining copies of the points in both the world and image coordinate systems.)
2. For each triangle Δ in the surface model.

- Compute a total radiation flux Φ_Δ for every triangle, according to equation (10), using world coordinates.

- Compute the area fractions f_Δ^p for all pixels p that are overlapped by the projection of the triangle by finding the geometrical intersection of Δ with those pixels.
- Update the pixel intensities \hat{I}_p based on equation (9) with the contribution from the current Δ .

3.1 Computation of the derivative matrix.

The inference of the surface model parameters depends on the ability to compute the derivatives of the observations with respect to the model parameters. In this section we obtain the derivatives of the pixel intensities with respect to the surface heights and albedos, and make some comments about efficient implementation.

According to equation (9), the intensity \hat{I}_p of a pixel p depends on the subset of the surface parameters, heights and albedos, that are associated with the triangles whose projections overlap the pixel area. It is seen from Eq. (10) that when the surface heights are fixed the radiation flux from a given triangle Φ_Δ depends linearly on the average albedos of the triangle, ρ_Δ . Then using Eqs. (9) and (5) the derivatives of a pixel intensity \hat{I}_p with respect to *logarithmically transformed* albedo values ρ'_Δ can be written in the following form:

$$\frac{\partial \hat{I}_p}{\partial \rho'_\Delta} = \rho_\Delta (1 - \rho_\Delta) f_\Delta^p \frac{\partial \Phi_\Delta}{\partial \rho_\Delta}. \quad (16)$$

where the form of the coefficient $\partial \Phi_\Delta / \partial \rho_\Delta$ immediately follows from Eq. (10). Note also that the derivative value in (16) is scaled with the area fraction factor for a given triangle.

3.1.1 Derivatives with respect to the heights of the vertices of the triangles

In our object-space renderer, which is based on pixel-triangle geometrical intersection in the image plane, the pixel intensity derivatives with respect to surface heights have two distinct contributions

$$\frac{\partial \hat{I}_p}{\partial z_i} = \sum_{\Delta} \left(f_\Delta^p \frac{\partial \Phi_\Delta}{\partial z_i} + \Phi_\Delta \frac{\partial f_\Delta^p}{\partial z_i} \right) \quad (17)$$

Variation of the surface height h_i gives rise to variations in the normals of the triangles associated with this height (in a general triangular mesh, on average 6 triangles are associated with each height) and this produces the derivatives of the total radiations fluxes Φ_Δ

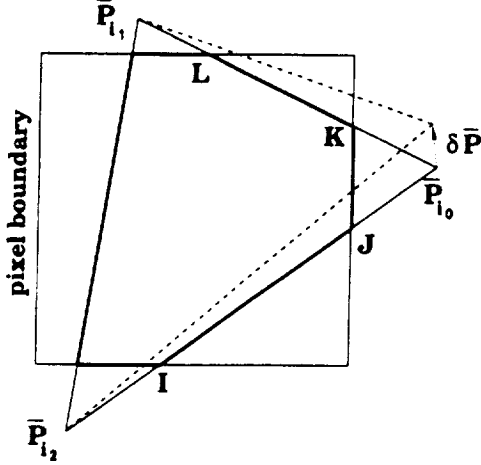


Figure 3: The intersection of the projection of a triangular surface element (i_0, i_1, i_2) onto the pixel plane with the pixel boundaries. Bold lines corresponds to the edges of the polygon resulting from the intersection. Dashed lines correspond to the new positions of the triangle edges when point P_{i_0} is displaced on $\delta \bar{\mathbf{P}}$

to the camera from those triangles. This is the first term in equation (17). Also, height variation gives rise to the displacement of the corresponding point which is the projection of this vertex on the image plane. This results in changes to the areas of the triangles and polygons with edges containing this point (see figure 3). This produces the derivatives of the fractions f_{Δ}^p , the second term in equation 17.

When the triangle is completely inside the pixel its area fraction $f_{\Delta}^p = 1$ and according to (17) its contribution to the pixel intensity derivative equals to the derivative of the corresponding radiation flux, $\partial \Phi / \partial z_1$. The flux derivative can be computed directly from the coordinates of the triangle vertices and the camera position using Eqs. (10) and (12). For the surface triangle with vertices $(P_{i_0}, P_{i_1}, P_{i_2})$ the flux derivative with respect to the z component of the vertex P_{i_0} equals

$$\frac{\partial \Phi}{\partial z_{i_0}} = \frac{1}{2} \rho (\mathbf{P}_{i_2} - \mathbf{P}_{i_1}) \times \hat{\mathbf{z}} \cdot \mathbf{g} \frac{S}{d^2}, \quad (18)$$

where

$$\mathbf{g} = \mathcal{I}_s (\hat{\mathbf{z}}_v \cos \alpha_s + \hat{\mathbf{z}}_s \cos \alpha_v - \hat{\mathbf{n}} \cos \alpha_s \cos \alpha_v) + \mathcal{I}_a \hat{\mathbf{z}}_v$$

and $\hat{\mathbf{z}}$ is a unit normal in the vertical direction.

When the triangle is projected into more than one pixel than the height derivatives of the projected area fraction in (17) have to be computed for every pixel

intersecting with the triangle. This can be done using the following chain rule arguments.

As mentioned above, when the z component of the vertex P_{i_0} in the 3D world coordinate system is varying by δz , the corresponding 2D point \bar{P}_{i_0} in the image projection plane is displaced by $\delta \bar{\mathbf{P}}$ as shown in figure 3. The corresponding point displacement derivative equals:

$$\frac{\partial \bar{\mathbf{P}}}{\partial z} = -\frac{1}{z_c} \begin{pmatrix} a f R_{13} + \bar{x} R_{33} \\ f R_{23} + \bar{y} R_{33} \end{pmatrix}. \quad (19)$$

Here we have dropped the vertex index; R_{ij} are the components of the rotation matrix $\bar{\mathbf{R}}$ in (13) and z_c is the z component of the point \mathbf{P} in the sensor coordinate system, given by equation (13).

Displacement $\delta \bar{\mathbf{P}}$ of the triangle vertex $\bar{\mathbf{P}}$ gives rise to the change $\delta \bar{A}_{\Delta}$ in the area of the projected triangle and also the change $\delta \bar{A}_{\text{polygon}}$ in the polygon area. It then follows from equation (15) that

$$\frac{\partial f_{\Delta}^p}{\partial z_{i_0}} = \frac{1}{\bar{A}_{\Delta}} \left(\frac{\partial \bar{A}_{\text{polygon}}}{\partial \bar{\mathbf{P}}_{i_0}} - f_{\Delta}^p \frac{\partial \bar{A}_{\Delta}}{\partial \bar{\mathbf{P}}_{i_0}} \right) \frac{\partial \bar{\mathbf{P}}_{i_0}}{\partial z_{i_0}}. \quad (20)$$

Here the point displacement derivative $\partial \bar{\mathbf{P}}_{i_0} / \partial z_{i_0}$ is given in (19).

Thus, the task of computing the derivative of the area fraction (20) is reduced to the computation of $\partial \bar{A}_{\Delta} / \partial \bar{\mathbf{P}}_{i_0}$ and $\partial \bar{A}_{\text{polygon}} / \partial \bar{\mathbf{P}}_{i_0}$. Note that the intersection of a triangle and a pixel for a rectangular pixel boundary can, in general, be a polygon with 3, 4, 5 or 6 edges with various possible forms. However the algorithm for computing the polygon area derivatives that we have developed is general, and does not depend on a particular polygon configuration. The main idea of the algorithm can be described as follows. Consider, as an example, the polygon shown in figure 3 which is a part of the projected surface triangle with indices i_0, i_1, i_2 . We are interested in the derivative of the polygon area with respect to the point $\bar{\mathbf{P}}_{i_0}$ that connects two edges of the projected triangle, (P_{i_2}, P_{i_0}) and (P_{i_0}, P_{i_1}) . These triangular edges contain segments (I, J) and (K, L) that are sides of the corresponding polygon. It can be seen from figure 3 that when the point $\bar{\mathbf{P}}_{i_0}$ is displaced by $\delta \bar{\mathbf{P}}_{i_0}$ the change in the polygon area is given by the sum of two terms

$$\delta \bar{A}_{\text{polygon}} = \delta A_{I,J} + \delta A_{K,L}$$

These terms are equal to the areas spanned by the two corresponding segments taken with appropriate signs. Therefore the polygon area derivative with the respect to the triangle vertex $\bar{\mathbf{P}}_{i_0}$ is represented as a

sum of the two "segment area" derivatives for the 2 segments adjacent to a given vertex. Using straightforward geometrical arguments one can calculate the areas $\delta A_{I,J}$ and $\delta A_{K,L}$ to first order in the displacement $\delta \bar{\mathbf{P}}_{i_0}$. Then the polygon area derivative can be expressed directly in terms of the triangle vertices and the endpoints of the polygon segments: I, J, K and L (cf. figure 3).

The details of the polygon area derivative computation will be presented elsewhere. Here we provide a result for the simplest particular case, i.e. the derivative of the area of the projected triangle

$$\frac{\partial \bar{A}_\Delta}{\partial \bar{\mathbf{P}}_{i_0}} = \frac{1}{2} \hat{\sigma} \cdot (\mathbf{P}_{i_1} - \mathbf{P}_{i_2}), \quad \hat{\sigma} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \quad (21)$$

The unit antisymmetrix matrix $\hat{\sigma}$ performs a $-\pi/2$ rotation in the image plane.

We note that the computation of the derivative matrix and the surface rendering essentially involves the same set of variables (triangle and polygon vertices, areas, etc). Therefore both computations can be done at the same time.

4 Adaptiveness and Super-resolution aspects

The correct choice of the smoothness prior (3) is very substantial for inferring the surfaces that have regions with high curvature (edges). It is also of especial importance in the case of surface super-resolution where the spatial resolution of individual pixels is greater than the size of a surface triangle.

Clearly that the values of metaparameters σ_h and σ_p should be controlled by the relative sizes of surface triangles ℓ and spatial resolution of individual pixels p . One needs at least 2 low-resolution images to infer heights and albedos at the same time. In this case σ_h and σ_p should be chosen in such a way to ensure the smoothness of a surface patches on a scale $\sim p$. For larger number of low-resolution images the inferred surface may have a smoothness scale smaller than p . This corresponds to a smaller values of σ_h and σ_p .

We note however that the smoothness prior remains very important even in the "over-constrained" case where the number of low-resolution images is relatively large and the total number of pixels in all images per height and albedo value is $\gtrsim 1$. The reason is that the spatial structure of the derivative values can be very irregular in this regime and smoothness prior essentially plays a role of *regularizer*. Indeed, one can show based on the analysis from the previous section that the magnitude of the derivatives $|\partial \hat{I}_p / \partial z_i|$ can be much larger for the surface vertices whose triangles intersect the pixel boundary than for the vertices that

are projected fully inside of the pixel along with all triangles surrounding them. When the number of triangles per pixel increases (typically > 10) this can give rise to a strong spatial modulation of the components \mathbf{b}_i of the gradient vector and also matrix elements of $\mathbf{D}\mathbf{D}^T$ (8). In general, pixel boundaries from different images are not alligned and therefore even for several images the spatial pattern of $\mathbf{b}_i \mathbf{D}\mathbf{D}^T$ and can be very irregular.

Regularization is achieved when correlations induced by the smoothness prior, Σ^{-1} , would be of the same order as correlations induced by observations in the matrix $\mathbf{D}\mathbf{D}^T$. Therefore every time we compute derivative matrix \mathbf{D} we readjust the metaparameter σ_h so that

$$\Gamma_{0,0} \sigma_p^2 / \sigma_h^2 \approx \text{Tr}_h (\mathbf{D}\mathbf{D}^T) / N \propto \ell^2, \quad (22)$$

where the traces of $\mathbf{D}\mathbf{D}^T$ are taken with respect to height and albedo variables; N is the number of vertices in a grid, ℓ is the size of individual triangles (equation 1). Value of σ_p is readjusted in a similar way.

Finally to achieve the local adaptiveness of the prior we place the curvature penalty on the **deviation** from the current surface estimate, $\mathbf{u} - \mathbf{u}_k$, but not on the estimate itself. In this case the second term in the expression the the gradient \mathbf{b} in (8) should be omitted.

5 Results

As a test example we used a triangulated surface which heights correspond to DEM of the Dackwater region (Nevada). We prepared surface albedos synthetically to fit the existing Landsat image data of the same region. The surface is of dimension 297×297 heights and the same number of albedos. Sixteen low-resolution images of the surface were produced using simple perspective sensor model (14), with differing lighting and camera orientations. Each images is 128×128 pixels. We used these synthetic images as the input data images I . Figure 4 (left) shows a portion of one of the input images of the size (40×40) pixels.

Starting from a mesh with all zero heights and all albedos set to 0.5, the conjugate gradient scheme described above was used to infer the surface shown in figure. The surface is of the same dimension as the original surface. Not that this is a dense triangulation – when projected into the pixel grid of figure 4 many triangles fall into one pixel. Thus we infer a super-resolved surface – a pixel lying on a mountain ridges does not imply a planar region in the inferred surface, rather, we infer a surface where highly curved regions

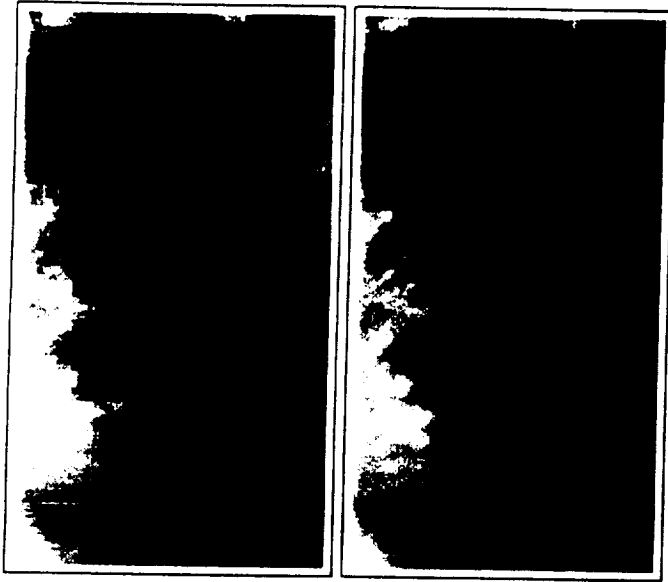


Figure 4: Left: a region of low-resolution input image, 96×74 pixels (left). Right: high-resolution reconstructed image, 960×740 pixels

may project into a single pixel. Figure 4 (right) shows a high-resolution image synthesized from the same region of the inferred surface as that corresponding to original data image at the left. This image was rendered at 10 times the resolution of the original data image. Comparison of left and right images figure 4 highlights the super-resolution aspect of our approach.

Because we know the original surface the error maps can be computed for both heights and albedos to judge the quality of inference. Note the vertical scales compared with figure . The reconstruction is accurate, with most errors being in the regions of high curvature.

6 Conclusions and future extensions

We have developed a very general framework for the inference of general surface geometry and reflectance models from image data, where the model choice is determined by the physical properties of the surface we wish to infer. We have demonstrated that for the case of a triangulated surface and Lambertian reflectance the parameters of a surface model, namely the heights and albedos, can be inferred from a set of image data. We have developed a framework that allows easy inclusion of future data observed from the same surface, and easy incorporation of data from other sensing modalities.

In this paper we assumed the registration parameters of input data images to be known in advance. In principle one can use the Bayesian approach devel-

oped above to infer the registration parameters of the data images along with surface heights and albedos. Such inference will include as an essential element the derivatives of the intensities of synthetic images with respect to registration parameters.

Another limitation of the current work is related to the absence of shadows and visible occlusions in input images. Future developments also include the addition of the ability to compute correctly both image and its derivatives when this limitation is lifted. Here we only note that the derivatives in the presence of shadows/occlusions are *nonlocal* as the points laying on the surface far apart can become correlated. This nonlocal derivatives are very informative as to the shape of the surface.

Among the other extensions are more realistic reflection functions, blurring and modeling of different surface topologies. Limits to the accuracy of the superresolved surface reconstruction will also be explored.

References

- [1] J. Bernardo and A. Smith. *Bayesian Theory*. Wiley, Chichester, New York, 1994.
- [2] J. Foley, A. van Dam, S. Finer, and J. Hughes. *Computer Graphics, principles and practice*. Addison-Wesley, 2nd ed. edition, 1990.
- [3] B. Horn and M. Brooks. *Shape from Shading*. MIT Press, 1989.
- [4] S. Nayar and M. Oren. Visual appearance of matte surfaces. *Science*, Vol. 267, pp. 1153–1156, 1995.
- [5] W. Rees. *Physical principles of remote sensing*. Cambridge University Press, 1990.
- [6] P. Cheesman, B. Kanefsky, R. Kraft, J. Stutz, and R. Hanson. Super-resolved surface reconstruction from multiple images. In G.R. Heidberg, editor, *Maximum Entropy and Bayesian Methods*, pp. 293–308, Kluwer, the Netherlands, 1996.
- [7] A. Patti, M. Sezan, and A. Tekalp. Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time. *IEEE Trans. Image Processing*, Vol. 6, pp. 1064–1076, 1997.
- [8] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Technical Report No 2273, INRIA, Sophia Antipolis, 1994.